Real-time Ground Estimation and Point Cloud Segmentation

Lukas N Rummelhard, Thomas Genevois, Jean-Alix David, Amaury Nègre, Christian Laugier

Real-time Ground Estimation and Point Cloud Segmentation

Lukas Rummelhard[1,2], Thomas Genevois[1], Jean-Alix David[1], Amaury Nègre[1,3] and Christian Laugier[1]
[1] Inria, Chroma, name.surname@inria.fr [2] currently CEA-LETI, name.surname@cea.fr
[3] currently Univ. Grenoble Alpes, CNRS, GIPSA-lab, name.surname@gipsa-lab.fr

Outline

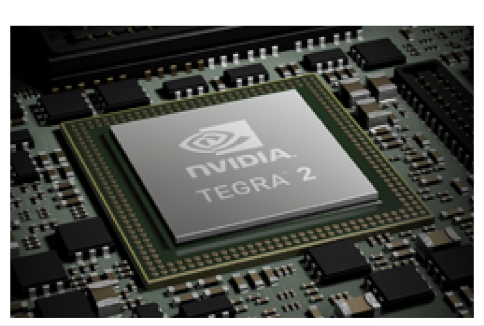
- A **generic ground form model**.
- Its **efficient estimation method**.
- Fully-integrated **ground-related point cloud segmentation**.
- **Parallelized process** on **GPU**.
- **Real-time** performances on **embedded devices (TX2)**.
- Applications to **ADAS** and **Autonomous Driving** frameworks
- **Validation** in **simulation** and **real experiments**

Context

- Development of low cost sensors and miniaturized computing units will enable advanced perception technologies to be embedded on realistic products.
- Many sensors provide 3D point clouds representing the geometry of the scene.
- Most perception solutions (object detection, occupancy grid generation, etc.) require a distinction between ground-related and obstacle-related points.
- This distinction necessitates a ground form estimation, complex enough to represent non-planar grounds, but efficient enough to provide real-time performances under the resource constraints of embedded devices.

Experimental Platform

- **Virtual platform** under **Gazebo**, with **realistic simulation** of motion, odometry and lidars.
- **Experimental platform on Renault Zoe**, equipped with Velodyne HDL64E, 4 Ibeo Lux LiDARs. Interconnection of sensors and programs using **ROS** middleware.
- Computations run on **Jetson Tegra X2**.



Ground Model

- **Spatio-Temporal Conditional Random Field**, sequence of regular planar nets of nodes N_i, to each node is associated a variable $G_i(h_i, sx_i, sy_i)^T$ representing the elevation and two directional slopes of the ground.
- Each data point (x_j, y_j, z_j) is associated to the closest node, C_j representing its belonging to the ground.
- Each G_i is related to potential functions.
  - **Measurement potential**:

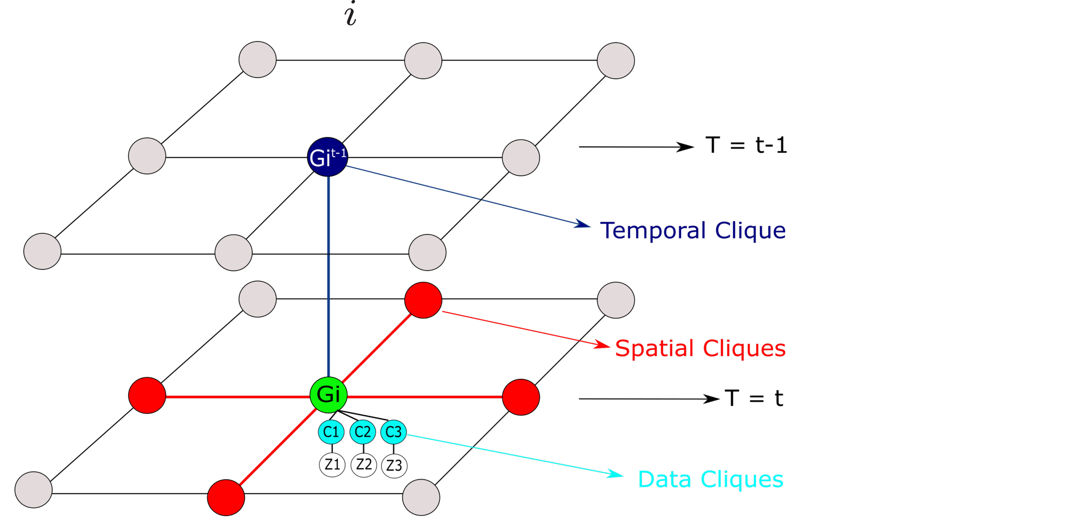
$$\psi \equiv exp(-\sum_i \alpha \sum_{j \in \mathcal{M}_i} c_j ||z_j - H_{ij} G_j||^2) \quad (1)$$

  - **Spatial potential**:

$$\Phi \equiv exp(-\sum_i \beta \sum_{j \in \mathcal{N}_i} ||G_i - F_{ij} G_j||^2) \quad (2)$$

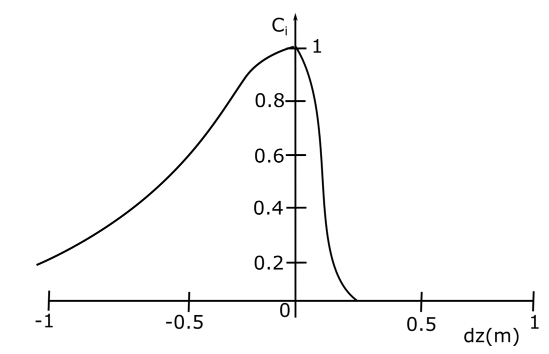
  - **Temporal potential**:

$$\xi \equiv exp(-\sum_i \gamma ||G_i - Q_i G_i^{t-1}||^2) \quad (3)$$



Model Inference

- G_i approximated to **Gaussian distributions**, leading to linear computations using information vectors and matrices (inverse of specified elements).
- **Expectation-Maximization**-like method, in parallel for every node.
  - **E step** : compute for each data point $P(C_i)$ to belong to the ground, depending on its height dz relative to the predicted ground elevation G_i



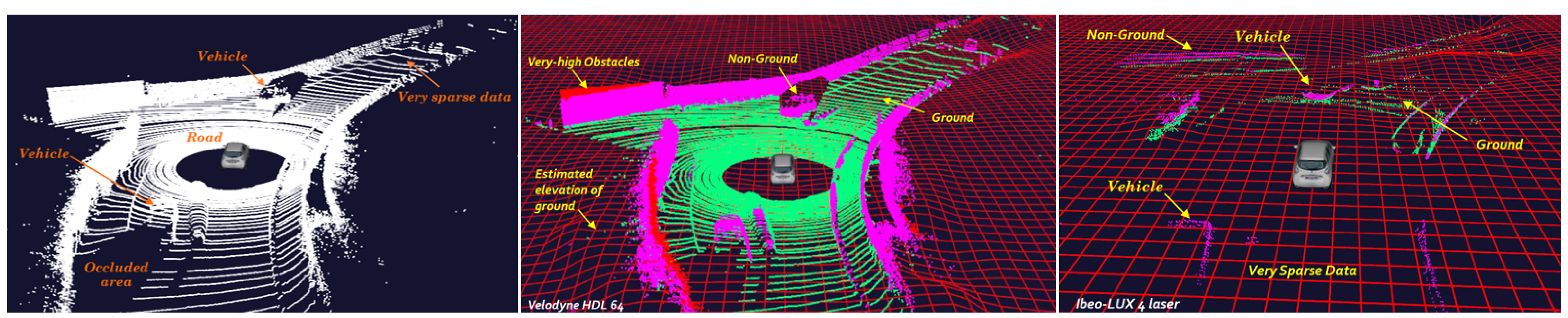
  - **M step** : compute G_i which minimizes the sum of the 3 potential functions (eq 1,2,3)

$$X_i^{k+1} = \alpha \sum_{j \in \mathcal{M}_i} c_j z_j H_{ij}^T + \beta \sum_{j \in \mathcal{N}_i} F_{ij}^T X_j^k + \gamma Q_i^T X_i^{t-1}$$

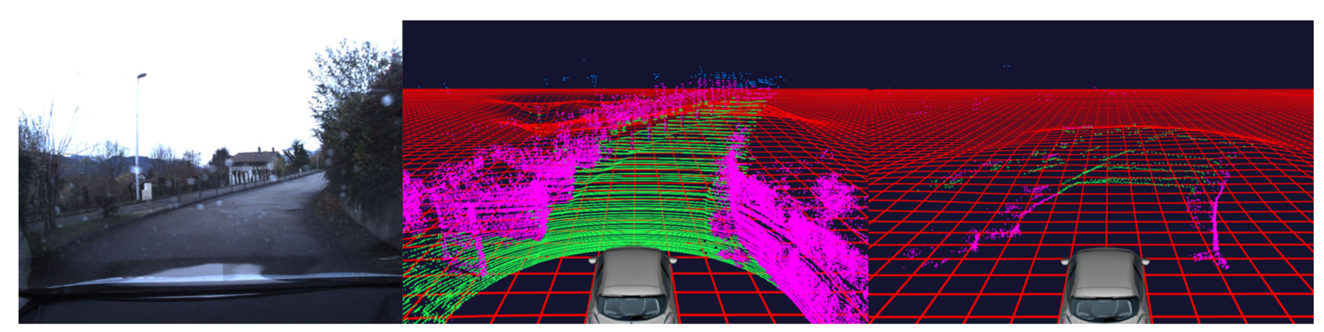
$$P_i^{k+1} = \alpha \sum_{j \in \mathcal{M}_i} c_j H_{ij}^T H_{ij} + \beta \sum_{j \in \mathcal{N}_i} F_{ij}^T P_j^k F_{ij} + \gamma Q_i^T P_i^{t-1} Q_i$$

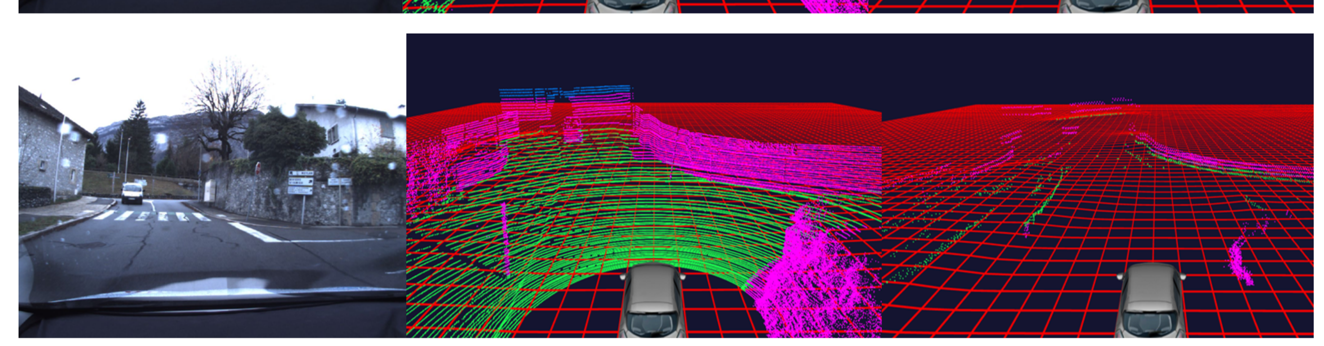
- Parallelized processing on GPU at every step, over sensor point cloud data in E steps, over ground elevation nodes in M steps. **Fully-integrated** in a global **GPU-based perception framework** (detailed in Applications)

Results



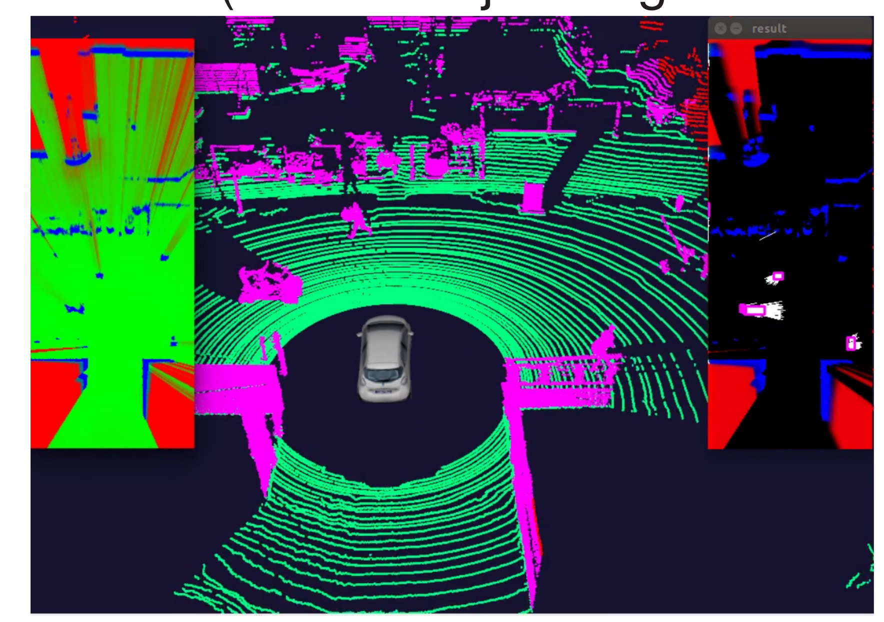
- Perception tested on **real road data** in various experimental environments (inner city, countryside, highways, mountain roads, etc.)
- **No prior knowledge** of the road
- With Velodyne or fused Ibeo point-cloud inputs
- **Real-time** ground estimation and data classification on **TX2**





Applications

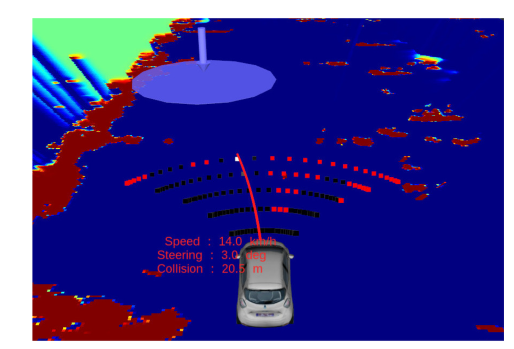
- Generation of **occupancy grids**, consistent with the classified point cloud and estimated ground model.
- Occupancy grid **filtering and tracking** over time, combined with **velocity inference**, at cell level (without object segmentation)



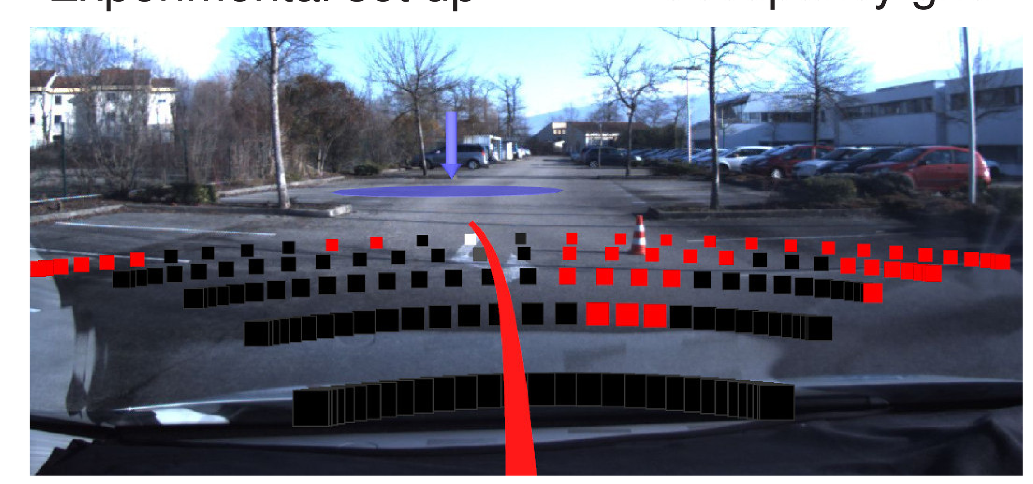
left : generated occupancy grid
right : filtered grid, with inferred velocity vectors

- **Grid-based collision risk** and time-to-collision estimation
- **Automatic emergency braking**, according to these potential collision estimates.
- **Autonomous driving**, by local pathfinding, using a **Dynamic Window Approach** (selection of the best trajectory in a set of online-computed trajectories).
- Validation in **simulation**, and **real experiments** with experimental platforms on a dedicated infrastructure, replicating realistic urban scenarios.

Experimental set up



Occupancy grid



Dynamic window and path leading to goal

Acknowledgements

This work has been developed within the scope of "Institut de Recherche Technologique" NanoElec, founded by the French program "Investissements d'Avenir" ANR-10-AIRT-05.